Verification and Synthesis of Timing Contracts for Embedded Controllers

Mohammad Al Khatib, Antoine Girard, Thao Dang

▶ **To cite this version:**

Mohammad Al Khatib, Antoine Girard, Thao Dang. Verification and Synthesis of Timing Contracts for Embedded Controllers. HSCC'16: Proceedings of the 19th international conference on hybrid systems: computation and control, Apr 2016, Vienna, Austria. pp.115-124, 10.1145/2883817.2883827 . hal-01276251

HAL Id: hal-01276251
https://hal.science/hal-01276251

Submitted on 19 Feb 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Verification and Synthesis of Timing Contracts for Embedded Controllers *

Mohammad Al Khatib
L2S, CNRS
CentraleSupélec
Université Paris-Sud
Université Paris-Saclay
F-91192 Gif-sur-Yvette
mohammad.alkhatib
@l2s.centralesupelec.fr

Antoine Girard
L2S, CNRS
CentraleSupélec
Université Paris-Sud
Université Paris-Saclay
F-91192 Gif-sur-Yvette
antoine.girard
@l2s.centralesupelec.fr

Thao Dang
Univ. Grenoble Alpes-CNRS
Verimag
F-38000 Grenoble
thao.dang@imag.fr

ABSTRACT

Timing contracts for embedded controller implementation specify the constraints on the time instants at which certain operations are performed such as sampling, actuation, computation, etc. In this paper, we consider the problem of verifying the stability of embedded control systems under such timing contracts. Reformulating the problem in the framework of impulsive linear systems, we provide theoretical conditions for stability and a verification algorithm based on reachability analysis. In the second part of the paper, given a model of the plant and of the controller we propose an approach to synthesize timing contracts that guarantee stability.

Categories and Subject Descriptors

J.7 [**Computer in other systems**]: Command and control, Real time

General Terms

Verification, Design, Algorithms

Keywords

Stability, Reachability, Impulsive linear systems, Sampled-data systems

1. INTRODUCTION

Physical systems equipped with embedded controllers are becoming pervasive (smart buildings, intelligent cars, drones, robots, etc.), thus increasing the need for high-confidence analysis and design tools that are able to handle tight interactions between the physical and digital worlds. In this context, contract-based approaches have been identified as a promising direction for cyber-physical systems design [28]. For instance, for embedded controller implementation, [11] proposed the use of timing contracts which specify the constraints on the time instants at which certain operations are performed such as sampling, actuation or computation. Under such contracts, the control engineers are responsible for designing a control law that is robust to all possible timing variation specified in the contract while the software engineers can focus on implementing the proposed control law so as to satisfy the timing contract. In this paper, we propose techniques that are useful within this framework. We first consider the problem of stability verification: given models of the physical plant and of the controller and a timing contract, verify that the resulting dynamical system is stable. We then tackle the problem of timing contract synthesis: given models of the physical plant and of the controller, determine a set of timing contracts that guarantee stability of the resulting system.

We adopt the impulsive linear dynamical system framework to model the overall system. Such systems form a class of hybrid systems which describe processes that evolve continuously and undergo instantaneous changes at discrete time instants. Applications of impulsive dynamical systems include sampled-data control systems [9], networked control systems [12], multi-agent systems [8], etc. In the present work, instantaneous changes occur at sampling and actuation times, t_k^s and t_k^a, $k \in \mathbb{N}$, which are assumed to be non-deterministic. More precisely, we assume that some uncertainty lies in each of the k^{th} sampling-to-actuation delay $\tau_k = t_k^a - t_k^s$ and sampling period $h_k = t_{k+1}^s - t_k^s$.

For the stability verification problem, we propose an approach based on the notion of reachable set. In the last decade, hybrid system reachability has had an important breakthrough in computing the reachable set corresponding to a linear continuous dynamics where the developed algorithms are based on representing the reachable sets by ellipsoids [22, 7], zonotopes [17, 2] or by support functions [24, 14]. In general, such algorithms handle time based switching by introducing auxiliary variables (clocks). In the following, we provide a specific approximation scheme for the reachable set at the sampling times to develop an effective stability verification approach. Then, we use this approach to tackle the timing contract synthesis problem. We propose a re-parametrization which provides some mono-

*This work was supported by the Agence Nationale de la Recherche (COMPACS project ANR-13-BS03-0004).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...$15.00.

tonicity property to the problem and allows us to develop an effective synthesis method based on guided sampling of the timing parameter space.

In all, we contribute in enriching our stability verification approach initiated in [1] by stating the proofs of the necessary and sufficient theoretical conditions for the stability of the impulsive linear system and by considering more complex timing contracts which require dedicated reachability algorithms for stability verification and more involved techniques for timing contract synthesis.

The paper is organized as follows. First, some preliminary notations are defined before formulating the stability verification and timing contract synthesis problems in Section 2. Stability conditions are provided in Section 3. Section 4 presents the reachable set approximation scheme and an algorithm for stability verification. In Section 5, we propose a solution to the timing contract synthesis problem. In Section 6, examples, some of which are used to compare our results with existing ones, are then discussed before concluding our work.

Notations. Let $\mathbb{R}$, $\mathbb{R}_0^+$, $\mathbb{R}^+$, $\mathbb{R}_0^-$, $\mathbb{R}^-$, $\mathbb{N}$, $\mathbb{N}^+$ denote the sets of reals, nonnegative reals, positive reals, nonpositive reals, negative reals, nonnegative integers and positive integers, respectively. For $I \subseteq \mathbb{R}_0^+$, let $\mathbb{N}_I = \mathbb{N} \cap I$. Given a real matrix $A \in \mathbb{R}^{n\times n}$, $|A|$ is the matrix whose elements are the absolute values of the elements of A. Given $\mathcal{S} \subseteq \mathbb{R}^n$ and a real matrix $A \in \mathbb{R}^{n\times n}$, the set $A\mathcal{S} = \{x \in \mathbb{R}^n : (\exists y \in \mathcal{S} : x = Ay)\}$; for $a \in \mathbb{R}$, $a\mathcal{S} = (aI_n)\mathcal{S}$ where I_n is the $n \times n$ identity matrix. The interior of $\mathcal{S}$ is denoted by $\text{int}(\mathcal{S})$. The convex hull of $\mathcal{S}$ is denoted by $\text{ch}(\mathcal{S})$. The interval hull of $\mathcal{S}$ is the smallest n-dimensional interval containing the set $\mathcal{S}$ and is denoted by $\square(\mathcal{S})$. The symmetric interval hull of $\mathcal{S}$ is the smallest symmetric (with respect to 0) n-dimensional interval containing $\mathcal{S}$ and is denoted by $\boxdot(\mathcal{S})$. Given $\mathcal{S}, \mathcal{S}' \subseteq \mathbb{R}^n$, the Minkowski sum of $\mathcal{S}$ and $\mathcal{S}'$ is $\mathcal{S} \oplus \mathcal{S}' = \{x + x' : x \in \mathcal{S}, x' \in \mathcal{S}'\}$. A polytope $\mathcal{P}$ is the intersection of a finite number of closed half-spaces, that is $\mathcal{P} = \{x \in \mathbb{R}^n : Hx \leq b\}$ where $H \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and the vector of inequalities is interpreted component-wise. Let H_i, $i \in \mathbb{N}_{[1,m]}$ denote the row vectors of H, then if $0 \in \text{int}(\text{ch}(\{H_1, \ldots, H_m\}))$, then $\mathcal{P}$ is compact. Given a template matrix $H \in \mathbb{R}^{m\times n}$ and a compact set $\mathcal{S} \subseteq \mathbb{R}^n$, let us define the polytope $\Gamma_H(\mathcal{S}) = \{x \in \mathbb{R}^n : Hx \leq b\}$ where $b_i = \max_{x\in\mathcal{S}} H_i x$, $i \in \mathbb{N}_{[1,m]}$. In other words, $\Gamma_H(\mathcal{S})$ is the smallest polytope whose facets directions are given by H and containing $\mathcal{S}$. We denote the set of all subsets of $\mathbb{R}^n$ by $2^{\mathbb{R}^n}$. We denote by $\mathcal{K}(\mathbb{R}^n)$ the set of compact subsets of $\mathbb{R}^n$ and by $\mathcal{K}_0(\mathbb{R}^n)$ the set of compact subsets of $\mathbb{R}^n$ containing 0 in their interior. For $x \in \mathbb{R}$, $\lceil x \rceil$ is the smallest integer not less than x, and for $c, c' \in \mathbb{R}^n$, $c \leq c'$ if and only if $c_i \leq c'_i$, $i = 1, \ldots, n$.

2. PROBLEM FORMULATION

2.1 Timing contracts for embedded control

In this work, we consider embedded control systems given under the form of general linear sampled-data control systems that take into account the sequences of sampling and actuation instants $(t_k^s)_{k\in\mathbb{N}}$ and $(t_k^a)_{k\in\mathbb{N}}$:

$$\begin{aligned} \dot{z}(t) &= Az(t) + Bu(t), \quad \forall t \in \mathbb{R}^+ \\ u(t) &= Kz(t_k^s), \qquad t_k^a < t \leq t_{k+1}^a \end{aligned} \tag{1}$$

where $z(t) \in \mathbb{R}^p$ is the state of the system, $u(t) \in \mathbb{R}^m$ is the control input, and $k \in \mathbb{N}$. For $t \in [0, t_0^a]$, $u(t)$ can be any constant value in $\mathbb{R}^m$.

We assume that the sequence of sampling and actuation instants (t_k^s) and (t_k^a) satisfy a *timing contract* given by

$$\begin{aligned} &t_0^s = 0,\ \tau_k = t_k^a - t_k^s \in [\underline{\tau}, \overline{\tau}], \\ &h_k = t_{k+1}^s - t_k^s \in [\max(\underline{h}, \tau_k), \overline{h}], \quad k \in \mathbb{N} \end{aligned} \tag{2}$$

where $\underline{\tau} \in \mathbb{R}_0^+$, $\overline{\tau} \in \mathbb{R}_0^+$, $\underline{h} \in \mathbb{R}^+$, and $\overline{h} \in \mathbb{R}^+$ provide bounds on the sampling-to-actuation delays (which includes time for computation of the control law) and sampling periods provided that $t_k^s \leq t_k^a \leq t_{k+1}^s$ for all $k \in \mathbb{N}$. Note that we impose $\underline{h} \neq 0$ to prevent Zeno behavior. Moreover, these parameters must belong to the following set so that the time intervals given in (2) are always non-empty:

$$\mathcal{C} = \left\{(\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}) \in \mathbb{R}_0^+ \times \mathbb{R}_0^+ \times \mathbb{R}^+ \times \mathbb{R}^+ : \underline{\tau} \leq \overline{\tau} \leq \overline{h}, \underline{h} \leq \overline{h}\right\}$$

Contract (2) is a general timing contract which includes or over-approximates the different contracts introduced in [11]. Their relation to the timing contract (2) is described as follows:

1. **ZET Contract**: The Zero Execution Time contract is given by (2) with $\underline{\tau} = \overline{\tau} = 0$ and $\underline{h} = \overline{h} = h \in \mathbb{R}^+$. In other words, the contract states that the sampling and actuation instants are periodic and simultaneous such that $t_k^s = t_k^a = kh$ for $k \in \mathbb{N}$. As mentioned in [11], this contract is hardly achievable in practice since computation always takes time in between the sampling and actuation instants.
2. **LET Contract**: The Logical Execution Time contract is given by (2) with $\underline{\tau} = \overline{\tau} = \underline{h} = \overline{h} = h \in \mathbb{R}^+$. The contract states that the sampling and actuation instants are periodic such that $t_0^s = 0$ and $t_k^s = t_{k-1}^a = kh$ for $k \in \mathbb{N}^+$.
3. **DET Contract**: The Deadline Execution Time contract is given by (2) with $\underline{\tau} = 0$ and $\underline{h} = \overline{h} = h \in \mathbb{R}^+$. The contract states that the sampling instants are periodic, or $t_k^s = kh$ for $k \in \mathbb{N}$, and actuation instants are at some point t_k^a in the interval $[t_k^s, t_k^s + \overline{\tau}]$, with $\overline{\tau} \leq h$.
4. **TOL Contract**: The Timing Tolerance contract is defined by a nominal sampling period $h \in \mathbb{R}^+$, nominal sampling to actuation delay $\tau \in \mathbb{R}_0^+$, and two jitters $J^h, J^\delta \in \mathbb{R}_0^+$ with $J^\tau \leq \tau$ and $J^h + J^\tau + \tau \leq h$, such that $t_k^s \in [kh, kh + J^h]$ and $t_k^a \in [t_k^s + \tau - J^\tau, t_k^s + \tau + J^\tau]$, for $k \in \mathbb{N}$ (refer to Figure 1). We cannot exactly model this contract using (2). However we can over-approximate it using (2) with $\underline{\tau} = \tau - J^\tau$, $\overline{\tau} = \tau + J^\tau$, $\underline{h} = h - J^h$, and $\overline{h} = h + J^h$. Thus stability of system (1) under this latter contract guarantees also its stability under the TOL contract.

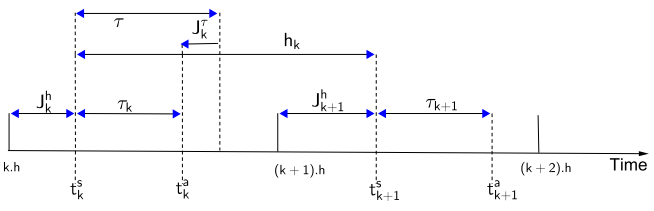

Figure 1: Time variables included in a *TOL* contract. $J_k^h \in [0, J^h]$ and $J_k^\tau \in [-J^\tau, J^\tau]$.

Table 1: Methods that can solve instances of Problem 1 with description of the modeling and computational approaches, list of restrictions and possible extensions.

	Models	Algorithm	Restrictions	Extensions
[10]	difference inclusion	LMI	–	$\tau_k > h_k$; controller synthesis
[12]		LMI	–	scheduling protocols
[20]		LMI	$\underline{\tau} = \overline{\tau} = 0$	controller synthesis
[21]		LMI	$\underline{\tau} = \overline{\tau} = 0$	–
[29]		SOS	$\underline{\tau} = \overline{\tau} = 0$	–
[13]		Invariant sets	$\underline{\tau} = \overline{\tau} = 0$	–
[1]		Reachability analysis	$\underline{\tau} = \overline{\tau} = 0$	stochastic timing uncertainty
[26]	time-delay systems	LMI	$\underline{h} = 0$	$\tau_k > h_k$; scheduling protocols
[16]		LMI	$\underline{h} = \overline{h}$, $\underline{\tau} = 0$	controller synthesis; quantization
[27]		LMI	$\underline{\tau} = \overline{\tau} = 0$	–
[15]		LMI	$\underline{h} = \underline{\tau} = \overline{\tau} = 0$	–
[4]	hybrid systems	SOS	–	nonlinear dynamics; scheduling protocols
[18]		LMI	$\underline{\tau} = 0$, $\underline{h} = 0$	scheduling protocols

It is noteworthy that system (1) has deterministic dynamics under any of the ZET or LET contracts, unlike the case of the DET or TOL contracts where at least one of the sampling-to-actuation delays or sampling period is time-varying. In our previous work [1] we considered the special case of nearly periodic linear impulsive systems (NPILS) which are modeled by (1) and (2) with $\underline{\tau} = \overline{\tau} = 0$. In other words, the sampling and actuation instants are simultaneous or $t_k^s = t_k^a$, and the duration in between two successive sampling instants is bounded in the interval $[\underline{h}, \overline{h}]$. Therefore, it is clear that we are dealing in this work with a more general timing contract which is more complex than the simple NPILS case.

2.2 Reformulation using impulsive systems

In our analysis it is more practical to transform equation (1) into an impulsive system with two types of resets each referring to a sampling or actuation instant. Such a reformulation is convenient to develop stability conditions based on reachability analysis. The system is thus given by:

$$\begin{aligned} \dot{x}(t) &= A_c x(t), t \neq t_k^s, t \neq t_k^a \\ x(t_k^{s+}) &= A_s x(t_k^s) \\ x(t_k^{a+}) &= A_a x(t_k^a) \end{aligned} \tag{3}$$

where $x(t) \in \mathbb{R}^n$ is the state of the system with $n = p + 2m$, (t_k^s) and (t_k^a) are given by (2), $x(t^+) = \lim_{\tau \to 0, \tau > 0} x(t+\tau)$, and

$$\begin{aligned} A_c &= \begin{pmatrix} A & 0 & B \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \; A_s = \begin{pmatrix} I_p & 0 & 0 \\ K & 0 & 0 \\ 0 & 0 & I_m \end{pmatrix}, \\ A_a &= \begin{pmatrix} I_p & 0 & 0 \\ 0 & I_m & 0 \\ 0 & I_m & 0 \end{pmatrix}, \; x(t) = \begin{pmatrix} z(t) \\ Kz(\theta^s(t)) \\ u(t) \end{pmatrix}, \end{aligned} \tag{4}$$

with $\theta^s(t) = t_k^s$ for $t \in (t_k^s, t_{k+1}^s]$.

In this paper, we consider stability in the following sense:

Definition 1. The system (2-3) is *globally uniformly exponentially stable* (GUES) if there exist $\lambda \in \mathbb{R}^+$ and $C \in \mathbb{R}^+$ such that, for all sequences $(t_k^s)_{k\in\mathbb{N}}$ and $(t_k^a)_{k\in\mathbb{N}}$ verifying (2) the solutions of (3) verify

$$\|x(t)\| \leq Ce^{-\lambda t} \|x(0)\|, \; \forall t \in \mathbb{R}^+.$$

We are now interested in verifying stability of embedded control systems in the form given by (1) under one of the general timing contracts defined previously. It is noteworthy that we can easily show that system (1) under the ZET and LET contracts is stable if and only if the eigenvalues of the matrix $e^{hA_c}A_aA_s$ and $A_ae^{hA_c}A_s$ are inside the unit circle respectively. As for the DET or TOL contracts, we have that stability of system (1) is guaranteed by the stability of (2-3) with an adequate choice of the timing contract parameters. Consequently, in this work, we consider the following problem:

PROBLEM 1 (STABILITY VERIFICATION). *Given A_c, A_s, $A_a \in \mathbb{R}^{n\times n}$, $(\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}) \in \mathcal{C}$, verify that (2-3) is GUES.*

This problem will be considered in the following section. Afterwards, we shall consider the problem of synthesizing timing contract parameters that guarantee the stability of the system. Given bounds on the parameters $0 \leq \tau_{min} \leq \tau_{max}$, $0 < h_{min} \leq h_{max}$, with $\tau_{min} \leq h_{min}$, $\tau_{max} \leq h_{max}$, let $\mathcal{D} = [\tau_{min}, \tau_{max}]^2 \times [h_{min}, h_{max}]^2$, the problem is formalized as follows:

PROBLEM 2 (TIMING CONTRACT SYNTHESIS). *Given A_c, A_s, $A_a \in \mathbb{R}^{n\times n}$ and $\mathcal{D}$, synthesize a set $\mathcal{C}^* \subseteq \mathcal{C} \cap \mathcal{D}$ such that for all $(\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}) \in \mathcal{C}^*$, (2-3) is GUES.*

Related work. Several approaches are developed in the literature to solve instances of Problem 1. A non-exhaustive list is given in Table 1. From the modeling perspective, the problem can be tackled using difference inclusions, time-delay systems or hybrid systems. On the computational side, most of the approaches are based on semi-definite programming using either Linear Matrix Inequalities (LMI) or Sum Of Squares (SOS) formulations. This makes a clear distinction with our approach which relies on reachability analysis. Let us remark that only a few approaches [10, 12, 4] appear to be able to address all instances of Problem 1. It is noticeable that [10, 12] have been implemented in the Networked Control Systems (NCS) toolbox [5] whose results will be compared to those of our approach. We should also acknowledge that some of these approaches are able to handle problems that we do not consider in the present work (possibility of having $\tau_k > h_k$, controller synthesis, scheduling protocols, quantization, nonlinear dynamics, stochastic

timing uncertainties). Finally, as far as we know, there is no available approach for addressing Problem 2 besides our preliminary work [1] where we impose $\underline{\tau} = \overline{\tau} = 0$.

3. STABILITY CONDITIONS

In this section we state necessary and sufficient theoretical conditions for system (2-3) to be GUES. In addition, we derive practical sufficient conditions that can be used to develop an algorithm for solving Problem 1.

3.1 Necessary and sufficient conditions

Our stability conditions are based on the notion of reachable set defined as follows:

Definition 2. Given a continuous-time dynamical system

$$\dot{x}(t) = Ax(t),\ t \in \mathbb{R}_0^+,\ x(t) \in \mathbb{R}^n$$

the *reachable set* on $[t, t'] \subseteq \mathbb{R}_0^+$ from the set $\mathcal{S} \subseteq \mathbb{R}^n$ is

$$\mathcal{R}^A_{[t,t']}(\mathcal{S}) = \bigcup_{\tau \in [t,t']} e^{\tau A}\mathcal{S}. \tag{5}$$

We also define the map: $\Phi : 2^{\mathbb{R}^n} \to 2^{\mathbb{R}^n}$, given for all $\mathcal{S} \subseteq \mathbb{R}^n$ by

$$\Phi(\mathcal{S}) = \bigcup_{\tau \in [\underline{\tau}, \overline{\tau}]} \bigcup_{w \in [\max(0, \underline{h} - \tau), \overline{h} - \tau]} e^{wA_c} A_a e^{\tau A_c} A_s \mathcal{S} \tag{6}$$

It is easy to see that if $\mathcal{S}$ is compact then so is $\Phi(\mathcal{S})$. It is clear that for two sets $\mathcal{S}$, $\mathcal{S}' \subseteq \mathbb{R}^n$ and $a \in \mathbb{R}$, we have $\Phi(\mathcal{S} \cup \mathcal{S}') = \Phi(\mathcal{S}) \cup \Phi(\mathcal{S}')$ and $\Phi(a\mathcal{S}) = a\Phi(\mathcal{S})$.

The interpretation of Φ is as follows. If $\mathcal{S}$ is the set of all states that are reachable by (2-3) at time t_k^s then $\Phi(\mathcal{S})$ is the set of reachable states at time t_{k+1}^s. We define the iterations of Φ as $\Phi^0(\mathcal{S}) = \mathcal{S}$ for all $\mathcal{S} \subseteq \mathbb{R}^n$, and $\Phi^{k+1} = \Phi \circ \Phi^k$ for all $k \in \mathbb{N}$. Then, for all $k \in \mathbb{N}$, $\Phi^k(\mathcal{S})$ is the set of reachable states by (2-3) at time t_k^s for initial states belonging to $\mathcal{S}$.

Next, we state the theoretical conditions on the stability of system (2-3) in terms of the map Φ. Similar results have been stated in [3] for discrete-time switched systems and in [1] for NPILS.

THEOREM 1. *Let $\mathcal{S} \in \mathcal{K}_0(\mathbb{R}^n)$, the following statements are equivalent:*

(a) System (2-3) is GUES,

(b) There exists a triplet $(k, j, \rho) \in \mathbb{N}^+ \times \mathbb{N}_{[0,k-1]} \times (0,1)$ such that $\Phi^k(\mathcal{S}) \subseteq \rho\Phi^j(\mathcal{S})$,

(c) There exists a pair $(k, \rho) \in \mathbb{N}^+ \times (0,1)$ such that $\Phi^k(\mathcal{S}) \subseteq \rho \bigcup_{j=0}^{k-1} \Phi^j(\mathcal{S})$.

PROOF. It is obvious that $(b) \implies (c)$. Hence, it is sufficient to prove that $(a) \implies (b)$ and $(c) \implies (a)$.

$(a) \implies (b)$: We will prove that there exists $(k, \rho) \in \mathbb{N}^+ \times (0,1)$ such that $\Phi^k(\mathcal{S}) \subseteq \rho\mathcal{S}$. This implies (b) with $j = 0$. Let $x(0) \in \mathcal{S}$, then $\Phi^k(\mathcal{S})$ represents all the possible values of $x(t_k^s)$. Since (2-3) is GUES, there exist $C \in \mathbb{R}^+$ and $\lambda \in \mathbb{R}^+$ such that

$$\|x(t_k^s)\| \leq Ce^{-\lambda t_k^s}\|x(0)\| \leq Ce^{-\lambda k\underline{h}}\|x(0)\|$$

which can be rewritten as

$$\Phi^k(\mathcal{S}) \subseteq Ce^{-\lambda k\underline{h}}\|x(0)\|\mathcal{B} \tag{7}$$

where $\mathcal{B}$ is the unit ball. Since $\mathcal{S} \in \mathcal{K}_0(\mathbb{R}^n)$, then there exist $\underline{c} \in \mathbb{R}^+$, $\overline{c} \in \mathbb{R}^+$ such that $\underline{c}\mathcal{B} \subseteq \mathcal{S} \subseteq \overline{c}\mathcal{B}$. Then, (7) and $x(0) \in \mathcal{S}$ give

$$\Phi^k(\mathcal{S}) \subseteq Ce^{-\lambda k\underline{h}}\overline{c}\mathcal{B} \subseteq \frac{Ce^{-\lambda k\underline{h}}\overline{c}}{\underline{c}}\mathcal{S}.$$

For k sufficiently large, $Ce^{-\lambda k\underline{h}}\overline{c} < \underline{c}$ and therefore (b) holds.

$(c) \implies (a)$: Let $\gamma = \rho^{\frac{1}{k}}$; since $\rho \in (0,1)$ then for all $j \in \mathbb{N}_{[0,k-1]}$, $\rho \leq \gamma^{k-j}$ and

$$\Phi^k(\mathcal{S}) \subseteq \rho \bigcup_{j=0}^{k-1} \Phi^j(\mathcal{S}) \subseteq \bigcup_{j=0}^{k-1} \gamma^{k-j}\Phi^j(\mathcal{S}). \tag{8}$$

Let $\mathcal{S}' = \bigcup_{j=0}^{k-1} \gamma^{-j}\Phi^j(\mathcal{S})$, then using properties of Φ:

$$\begin{aligned}\Phi(\mathcal{S}') &= \Phi\left(\bigcup_{j=0}^{k-1} \gamma^{-j}\Phi^j(\mathcal{S})\right) = \bigcup_{j=0}^{k-1} \gamma^{-j}\Phi^{j+1}(\mathcal{S}) \\ &= \left(\bigcup_{j=0}^{k-2} \gamma^{-j}\Phi^{j+1}(\mathcal{S})\right) \cup \gamma^{-k+1}\Phi^k(\mathcal{S})\end{aligned}$$

Making a change of index in the union and using (8) yield

$$\begin{aligned}\Phi(\mathcal{S}') &\subseteq \left(\bigcup_{j=1}^{k-1} \gamma^{-j+1}\Phi^j(\mathcal{S})\right) \cup \gamma^{-k+1}\left(\bigcup_{j=0}^{k-1} \gamma^{k-j}\Phi^j(\mathcal{S})\right) \\ &\subseteq \gamma\left(\bigcup_{j=0}^{k-1} \gamma^{-j}\Phi^j(\mathcal{S})\right) = \gamma\mathcal{S}'. \end{aligned} \tag{9}$$

Since $\mathcal{S} \in \mathcal{K}_0(\mathbb{R}^n)$, then $\mathcal{S}'$ is compact. Moreover, since it contains $\mathcal{S}$, then $\mathcal{S}' \in \mathcal{K}_0(\mathbb{R}^n)$. Then, there exist $\underline{c}' \in \mathbb{R}^+$, $\overline{c}' \in \mathbb{R}^+$ such that $\underline{c}'\mathcal{B} \subseteq \mathcal{S}' \subseteq \overline{c}'\mathcal{B}$. Now consider a trajectory x of (2-3), then $x(0) \in \|x(0)\|\mathcal{B} \subseteq \frac{\|x(0)\|}{\underline{c}'}\mathcal{S}'$ and (9) gives for all $i \in \mathbb{N}$

$$x(t_i^s) \in \Phi^i\left(\frac{\|x(0)\|}{\underline{c}'}\mathcal{S}'\right) \subseteq \frac{\|x(0)\|\gamma^i}{\underline{c}'}\mathcal{S}' \subseteq \frac{\|x(0)\|\gamma^i\overline{c}'}{\underline{c}'}\mathcal{B}.$$

In other words, it holds for all $i \in \mathbb{N}$,

$$\|x(t_i^s)\| \leq \frac{\gamma^i\overline{c}'}{\underline{c}'}\|x(0)\|.$$

Now, let $t \in \mathbb{R}^+$, let $i \in \mathbb{N}$ be such that $t \in (t_i^s, t_{i+1}^s]$, then $t - t_i^s \leq \overline{h}$. Moreover, if $t \in (t_i^s, t_i^a]$, then

$$\|x(t)\| \leq e^{\|A_c\|\overline{h}}\|A_s\|\frac{\gamma^i\overline{c}'}{\underline{c}'}\|x(0)\|$$

and if $t \in (t_i^a, t_{i+1}^s]$, then

$$\|x(t)\| \leq e^{\|A_c\|\overline{h}}\|A_a\|\|A_s\|\frac{\gamma^i\overline{c}'}{\underline{c}'}\|x(0)\|.$$

In addition, we have $i \geq t/\overline{h}$ and since $\gamma \in (0,1)$ it follows that for all $t \in \mathbb{R}^+$

$$\begin{aligned}\|x(t)\| &\leq \frac{e^{\|A_c\|\overline{h}}\max(\|A_a\|, 1)\|A_s\|\overline{c}'}{\underline{c}'}\gamma^{(t/\overline{h})}\|x(0)\| \\ &\leq \frac{e^{\|A_c\|\overline{h}}\max(\|A_a\|, 1)\|A_s\|\overline{c}'}{\underline{c}'}e^{\frac{\ln(\gamma)}{\overline{h}}t}\|x(0)\|.\end{aligned}$$

Since $\gamma \in (0,1)$, (2-3) is GUES. $\square$

3.2 Sufficient conditions

The map Φ involved in Theorem 1 is in general impossible to compute exactly. Then, we may use an over-approximation $\overline{\Phi} : \mathcal{K}(\mathbb{R}^n) \to \mathcal{K}(\mathbb{R}^n)$ satisfying the following assumption:

ASSUMPTION 1. *For all $\mathcal{S} \in \mathcal{K}(\mathbb{R}^n)$, $\Phi(\mathcal{S}) \subseteq \overline{\Phi}(\mathcal{S})$.*

We compute the map $\overline{\Phi}$ instead of Φ in order to derive the practical condition on stability used in the stability verification algorithm later on to solve Problem 1. Section 4 discusses on the effective computation of the map $\overline{\Phi}$. We now derive sufficient conditions for stability based on $\overline{\Phi}$.

COROLLARY 1. *Under Assumption 1, if there exist a set $\mathcal{S} \in \mathcal{K}_0(\mathbb{R}^n)$ and a triplet $(k, i, \rho) \in \mathbb{N}^+ \times \mathbb{N}_{[0,k-1]} \times (0, 1)$ such that $\overline{\Phi}^k(\mathcal{S}) \subseteq \rho\overline{\Phi}^i(\mathcal{S})$, then system (2-3) is GUES.*

PROOF. $\overline{\Phi}^k(\mathcal{S}) \subseteq \rho\overline{\Phi}^i(\mathcal{S}) \subseteq \rho\bigcup_{j=0}^{k-1}\overline{\Phi}^j(\mathcal{S})$. Then similar to the second part of the proof of Theorem 1, let $\mathcal{S}' = \bigcup_{j=0}^{k-1}\gamma^{-j}\overline{\Phi}^j(\mathcal{S})$ where $\gamma = \rho^{\frac{1}{k}}$. Then

$$\begin{aligned}\Phi(\mathcal{S}') &= \Phi\left(\bigcup_{j=0}^{k-1}\gamma^{-j}\overline{\Phi}^j(\mathcal{S})\right) = \bigcup_{j=0}^{k-1}\gamma^{-j}\Phi(\overline{\Phi}^j(\mathcal{S})) \\ &\subseteq \bigcup_{j=0}^{k-1}\gamma^{-j}\overline{\Phi}(\overline{\Phi}^j(\mathcal{S})) = \bigcup_{j=0}^{k-1}\gamma^{-j}\overline{\Phi}^{j+1}(\mathcal{S}).\end{aligned}$$

Then, following the same steps as in (9), we can show that $\Phi(\mathcal{S}') \subseteq \gamma\mathcal{S}'$. Following the same line as in the proof of Theorem 1, one concludes that (2-3) is GUES. □

The previous corollary provides the background for designing a solution to Problem 1 in the next section.

4. OVER-APPROXIMATION SCHEME AND STABILITY VERIFICATION

In this section, we present an approach for computing an over-approximation of Φ. Furthermore, we develop an algorithm providing a solution to Problem 1.

4.1 Over-approximation

We first state the following result from [23] which gives an over-approximation scheme for the reachable set given by (5).

THEOREM 2. *[23] Let $T \in \mathbb{R}^+$, $A \in \mathbb{R}^{n\times n}$, $\mathcal{S} \in \mathcal{K}(\mathbb{R}^n)$ and $N \in \mathbb{N}^+$, let*

$$\overline{\mathcal{R}}^A_{[0,T]}(\mathcal{S}) = \bigcup_{i=1}^{N}\overline{\mathcal{R}}^A_{[(i-1)\delta,i\delta]}(\mathcal{S})$$

where $\delta = T/N$ is the time step, and $\overline{\mathcal{R}}^A_{[(i-1)\delta,i\delta]}(\mathcal{S})$ is defined by the recurrence equation:

$$\begin{aligned}\overline{\mathcal{R}}^A_{[0,\delta]}(\mathcal{S}) &= ch(\mathcal{S}, e^{\delta A}\,\mathcal{S}) \,\oplus\, 1/4\;\epsilon_\delta(\mathcal{S}), \qquad (10)\\ \overline{\mathcal{R}}^A_{[i\delta,(i+1)\delta]}(\mathcal{S}) &= e^{\delta A}\,\overline{\mathcal{R}}^A_{[(i-1)\delta,i\delta]}(\mathcal{S}),\; i \in \mathbb{N}_{[1,N-1]}\end{aligned}$$

with

$$\begin{aligned}\epsilon_\delta(\mathcal{S}) = {} & \square(|A|^{-1}(e^{\delta|A|} - I) \boxdot (A(I - e^{\delta A})\mathcal{S})) \oplus \\ & \square(|A|^{-2}(e^{\delta|A|} - I - \delta|A|) \boxdot (A^2 e^{\delta A}\mathcal{S})).\end{aligned}$$

Then, $\mathcal{R}^A_{[(i-1)\delta,i\delta]}(\mathcal{S}) \subseteq \overline{\mathcal{R}}^A_{[(i-1)\delta,i\delta]}(\mathcal{S})$, for all $i \in \mathbb{N}_{[1,N]}$ and $\mathcal{R}^A_{[0,T]}(\mathcal{S}) \subseteq \overline{\mathcal{R}}^A_{[0,T]}(\mathcal{S})$.

The previous theorem can serve to compute an over-approximation of Φ. Indeed, from (6), one can easily check that

$$\begin{aligned}\Phi(\mathcal{S}) &\subseteq \mathcal{R}^{A_c}_{[\max(0,\underline{h}-\overline{\tau}),\overline{h}-\underline{\tau}]}\left(A_a\mathcal{R}^{A_c}_{[\underline{\tau},\overline{\tau}]}(A_s\mathcal{S})\right)\\ &\subseteq e^{\max(0,\underline{h}-\overline{\tau})A_c}\mathcal{R}^{A_c}_{[0,\min(\overline{h}-\underline{\tau},\overline{h}-\underline{h}+\overline{\tau}-\underline{\tau})]}\Big(\\ &\qquad A_a e^{\underline{\tau}A_c}\mathcal{R}^{A_c}_{[0,\overline{\tau}-\underline{\tau}]}(A_s\mathcal{S})\Big) \qquad (11)\end{aligned}$$

with in turn can easily be over-approximated using the result of Theorem 2. In the case of NPILS, the previous inclusion becomes an equality. This is the approach followed in our previous work [1]. However, for the general timing contract (2), the coupling in the timing uncertainties w and τ in (6) is totally disregarded in (11) and therefore leads to conservatism. Therefore, in this paper, to reduce conservatism, we present a specific approximation scheme for Φ, that takes into consideration the coupling in the timing uncertainties. It is based on the following result:

LEMMA 1. *Let $\mathcal{S} \in \mathcal{K}(\mathbb{R}^n)$, let $N_1, N_2 \in \mathbb{N}^+$, then*

$$\begin{aligned}\Phi(\mathcal{S}) \subseteq {} & \bigcup_{j_1=1}^{N_1}\bigcup_{j_2=1}^{n_2(j_1)} e^{(\theta(j_1)+(j_2-1)\delta_2)A_c}\mathcal{R}^{A_c}_{[0,\delta_2]}\Big(\\ & A_a e^{(\underline{\tau}+(j_1-1)\delta_1)A_c}\mathcal{R}^{A_c}_{[0,\delta_1]}(A_s\mathcal{S})\Big) \qquad (12)\end{aligned}$$

where for $j_1 \in \mathbb{N}_{[1,N_1]}$

$$\begin{aligned}\delta_1 &= (\overline{\tau} - \underline{\tau})/N_1\\ \delta_2 &= \min(\overline{h} - \underline{\tau}, \overline{h} - \underline{h} + \delta_1)/N_2\\ \theta(j_1) &= \max(0, \underline{h} - \underline{\tau} - j_1\delta_1)\\ n_2(j_1) &= \lceil\min(\overline{h} - \underline{\tau} - (j_1 - 1)\delta_1, \overline{h} - \underline{h} + \delta_1)/\delta_2\rceil.\end{aligned} \qquad (13)$$

PROOF. From (6), it follows that

$$\begin{aligned}\Phi(\mathcal{S}) = {} & \bigcup_{j_1=1}^{N_1}\bigcup_{\tau\in[\underline{\tau}+(j_1-1)\delta_1,\underline{\tau}+j_1\delta_1]}\\ & \bigcup_{w\in[\max(0,\underline{h}-\tau),\overline{h}-\tau]} e^{wA_c}A_a e^{\tau A_c}A_s\mathcal{S}\\ \subseteq {} & \bigcup_{j_1=1}^{N_1}\mathcal{R}^{A_c}_{[\theta(j_1),\overline{h}-\underline{\tau}-(j_1-1)\delta_1]}\Big(\\ & A_a\mathcal{R}^{A_c}_{[\underline{\tau}+(j_1-1)\delta_1,\underline{\tau}+j_1\delta_1]}(A_s\mathcal{S})\Big)\\ \subseteq {} & \bigcup_{j_1=1}^{N_1} e^{\theta(j_1)A_c}\mathcal{R}^{A_c}_{[0,\overline{h}-\underline{\tau}-(j_1-1)\delta_1-\theta(j_1)]}\Big(\\ & A_a e^{(\underline{\tau}+(j_1-1)\delta_1)A_c}\mathcal{R}^{A_c}_{[0,\delta_1]}(A_s\mathcal{S})\Big).\end{aligned}$$

Remarking that

$$\overline{h} - \underline{\tau} - (j_1 - 1)\delta_1 - \theta(j_1) = \min(\overline{h} - \underline{\tau} - (j_1 - 1)\delta_1, \overline{h} - \underline{h} + \delta_1)$$

one gets

$$\begin{aligned}\Phi(\mathcal{S}) \subseteq {} & \bigcup_{j_1=1}^{N_1}\bigcup_{j_2=1}^{n_2(j_1)} e^{\theta(j_1)A_c}\mathcal{R}^{A_c}_{[(j_2-1)\delta_2,j_2\delta_2]}\Big(\\ & A_a e^{(\underline{\tau}+(j_1-1)\delta_1)A_c}\mathcal{R}^{A_c}_{[0,\delta_1]}(A_s\mathcal{S})\Big)\end{aligned}$$

which leads to (12). □

REMARK 1. *N_1 and N_2 are parameters used to discretize time intervals. For $N_1 = N_2 = 1$, the over-approximation given by (12) is the same as the one in (11).*

We now present our over-approximation scheme for Φ:

THEOREM 3. *Let $\mathcal{S} \in \mathcal{K}(\mathbb{R}^n)$, $N_1, N_2 \in \mathbb{N}^+$, and $H \in \mathbb{R}^{m\times n}$, such that $0 \in int(ch(\{H_1, \ldots, H_m\}))$, let $\overline{\Phi} : \mathcal{K}(\mathbb{R}^n) \to \mathcal{K}(\mathbb{R}^n)$ be given by*

$$\overline{\Phi}(\mathcal{S}) = \Gamma_H \left(\bigcup_{j_1=1}^{N_1} \bigcup_{j_2=1}^{n_2(j_1)} e^{(\theta(j_1)+(j_2-1)\delta_2)A_c} \overline{\Phi}_{j_1}(\mathcal{S}) \right)$$

where for $j_1 \in \mathbb{N}_{[1,N_1]}$,

$$\overline{\Phi}_{j_1}(\mathcal{S}) = \overline{\mathcal{R}}^{A_c}_{[0,\delta_2]} \left(A_a e^{(\underline{\tau}+(j_1-1)\delta_1)A_c} \overline{\mathcal{R}}^{A_c}_{[0,\delta_1]}(A_s\mathcal{S}) \right)$$

with δ_1, δ_2, $\theta(j_1)$, $n_2(j_1)$ given by (13) , and $\overline{\mathcal{R}}^{A_c}_{[0,\delta_1]}$, $\overline{\mathcal{R}}^{A_c}_{[0,\delta_2]}$ computed as in (10). Then, $\Phi(\mathcal{S}) \subseteq \overline{\Phi}(\mathcal{S})$.

PROOF. The proof is straightforward from Theorem 2 and Lemma 1. □

REMARK 2. *In the previous result, the operation Γ_H is not necessary to guarantee over-approximation of Φ. On the other hand, without this operation, the over-approximation of Φ would be given by the union of possibly numerous sets which may be quite impractical for subsequent manipulations. For that reason, this union is over-approximated by the smallest enclosing polytope whose facets direction are given by a matrix H. Moreover, if $\mathcal{S}$ is a polytope, then using the properties of support functions [24], the computation of $\overline{\Phi}(\mathcal{S})$ reduces to solving a set of linear programs.*

We illustrate the tightness of our new approximation scheme using system (16) (see Section 6) with the timing contract given by $\underline{\tau} = 0$, $\overline{\tau} = 0.4$, $\underline{h} = 0.2$, $\overline{h} = 1.2$. We consider a polytope $\mathcal{S}$ defined by a matrix H with 44 rows. Figure 2 shows sampled points (in grey) from $\Phi(\mathcal{S})$. The white polytope corresponds to the over-approximation $\overline{\Phi}(\mathcal{S})$ given in Theorem 3 with $N_1 = 20$ and $N_2 = 50$. The black polytope is given by (11) over-approximated using Theorem 2 where $\overline{\mathcal{R}}^{A_c}_{[0,\overline{\tau}-\underline{\tau}]}$ and $\overline{\mathcal{R}}^{A_c}_{[0,\min(\overline{h}-\underline{\tau},\overline{h}-\underline{h}+\overline{\tau}-\underline{\tau})]}$ are computed with $N = 20$ and $N = 50$ respectively. One can check that the over-approximation given by Theorem 3 is quite tight and much less conservative than that given by (11).

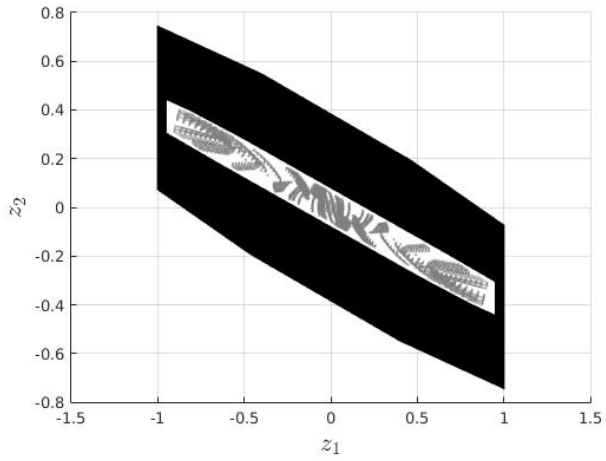

Figure 2: Sampled points of $\Phi(\mathcal{S})$ (in grey), over-approximation $\overline{\Phi}(\mathcal{S})$ given by Theorem 3 and over-approximation of (11) computed using Theorem 2 (in black).

4.2 Stability verification algorithm

We now present our stability verification algorithm which consists of two main steps: an initialization step where an initial set $\mathcal{P}_0$ is computed and a main loop which tries to verify the sufficient stability condition given in Corollary 1 by iterating the map $\overline{\Phi}$ given by Theorem 3 from the set $\mathcal{P}_0$.

4.2.1 Initialization

The choice of the initial set $\mathcal{P}_0$ is crucial as it may impact significantly the number of iterations of $\overline{\Phi}$ that are necessary to check the condition of Corollary 1. Intuitively, in order to minimize this number of iterations, $\mathcal{P}_0$ should be already close to an invariant set. Indeed, if $\overline{\Phi}(\mathcal{P}_0) \subseteq \mathcal{P}_0$, the stability condition holds after only one iterate of $\overline{\Phi}$. One way to choose $\mathcal{P}_0$ close to an invariant set is to define $\mathcal{P}_0$ as a common contracting polytope to $L \in \mathbb{N}^+$ linear discrete-time systems, such that

$$\forall j \in \mathbb{N}_{[1,L]},\ e^{(h_j-\tau_j)A_c} A_a e^{\tau_j A_c} A_s \mathcal{P}_0 \subseteq \text{int}(\mathcal{P}_0),$$

where the couples (τ_j, h_j) satisfy timing contract (2) for all $j \in \mathbb{N}_{[1,L]}$. Then, $\mathcal{P}_0$ can be computed either using a backward iterative method as in [6] and [13] or using a forward iterative method as in [3]. We denote the function computing $\mathcal{P}_0$ by $\mathsf{init}(A_c, A_d, A_s, \underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}, L)$. Then, $\mathcal{P}_0 = \{x \in \mathbb{R}^n : Hx \leq b_0\}$. The matrix H defining $\mathcal{P}_0$ is used in the main loop of the algorithm in the computation of the map $\overline{\Phi}$.

4.2.2 Main loop

The initial set is propagated using the map $\overline{\Phi}$ given by Theorem 3. Then if the stability condition given by Corollary 1 is verified, system (2-3) is GUES and the algorithm returns true. Otherwise, if a maximum number of iterations, k_{max}, is reached then the algorithm fails to prove stability and returns unknown. The algorithm that solves Problem 1 is given as follows:

ALGORITHM 1. *Stability verification*

function *is_GUES(A_c,A_a,A_s,$\underline{\tau}$,$\overline{\tau}$,$\underline{h}$,$\overline{h}$)*
input: *A_c, A_a, $A_s \in \mathbb{R}^{n\times n}$, $(\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}) \in \mathcal{C}$*
output: *true if system (2-3) is proved GUES, unknown otherwise*
parameter: *N_1,N_2,L, $k_{max} \in \mathbb{N}^+$*

1: $\mathcal{P}_0$:=*init*$(A_c, A_d, A_s, \underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}, L)$; ▷ *compute initial set*
2: ***for*** $k = 1$ *to* k_{max} ***do***
3: $\quad\mathcal{P}_k$:=$\overline{\Phi}(\mathcal{P}_{k-1})$; ▷ *set propagation*
4: $\quad$***if*** $\exists i \in \mathbb{N}_{[0,k-1]}$, $\mathcal{P}_k \subseteq int(\mathcal{P}_i)$ ***then***
5: $\quad\quad$*return true;* ▷ *system (2-3) is GUES*
6: $\quad$***end if***
7: ***end for***
8: *return unknown;*

Note that all polytopes $\mathcal{P}_k$ are of the form $\mathcal{P}_k = \{x \in \mathbb{R}^n : Hx \leq b_k\}$, then the inclusion test at line 4 only consists in checking $b_k \leq b_i$. Although Algorithm 1 is only based on sufficient conditions for the stability of system (2-3), its effectiveness will be demonstrated on numerical examples in Section 6.

5. TIMING CONTRACT SYNTHESIS

In this section, we propose a solution to Problem 2. We first define a re-parametrization of the timing-contract such that stability of system (2-3) becomes monotone with respect to the new parameters. Monotonicity is a very attractive property for designing efficient heuristics for timing contract synthesis since stability is preserved when the parameter values increase. This allows us to tackle the timing contract synthesis by sampling the parameter space.

5.1 Re-parametrization

Let us denote the vector of timing contract parameters $\alpha = (\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}) \in \mathcal{D} = [\tau_{min}, \tau_{max}]^2 \times [h_{min}, h_{max}]^2$, where $0 \leq \tau_{min} \leq \tau_{max}$, $0 < h_{min} \leq h_{max}$, $\tau_{min} \leq h_{min}$, $\tau_{max} \leq h_{max}$. For $\alpha \in \mathcal{C} \cap \mathcal{D}$ we denote the property:

$$\mathsf{Stab}(\alpha) \equiv \text{(2-3) is GUES with parameters } \alpha.$$

Solving Problem 2 is equivalent to computing (a subset of) the set $\mathcal{C}_o$ defined by

$$\mathcal{C}_o = \{\alpha \in \mathcal{C} \cap \mathcal{D} : \mathsf{Stab}(\alpha)\}.$$

Let us define a new parameter $\beta = (\beta_1, \beta_2, \beta_3, \beta_4) \in \mathcal{D}'$ where $\mathcal{D}' = [\tau_{min}, \tau_{max}] \times [-\tau_{max}, -\tau_{min}] \times [h_{min}, h_{max}] \times [-h_{max}, -h_{min}]$ and the map $f : \mathcal{D}' \to \mathcal{D}$ such that $f(\beta) = \alpha = (\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h})$ where

$$\underline{\tau} = \beta_1,\ \overline{\tau} = \min(-\beta_2, -\beta_4),\ \underline{h} = \beta_3,\ \overline{h} = -\beta_4.$$

We define the following constraint set for the parameter β:

$$\mathcal{C}' = \left\{ \beta \in \mathbb{R}_0^+ \times \mathbb{R}_0^- \times \mathbb{R}^+ \times \mathbb{R}^- : \begin{array}{l} \beta_1 \leq \min(-\beta_2, -\beta_4) \\ \beta_3 \leq -\beta_4 \end{array} \right\}.$$

The following result holds:

LEMMA 2. *Let $\mathcal{C}'_o$ be given by*

$$\mathcal{C}'_o = \{\beta \in \mathcal{C}' \cap \mathcal{D}' : \mathit{Stab}(f(\beta))\}.$$

Then, $f(\mathcal{C}' \cap \mathcal{D}') = \mathcal{C} \cap \mathcal{D}$ and $f(\mathcal{C}'_o) = \mathcal{C}_o$.

PROOF. Let us first show that $f(\mathcal{C}' \cap \mathcal{D}') \subseteq \mathcal{C} \cap \mathcal{D}$ and $f(\mathcal{C}'_o) \subseteq \mathcal{C}_o$. Let $\beta \in \mathcal{C}' \cap \mathcal{D}'$ and $\alpha = f(\beta) = (\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h})$. Then, $\beta \in \mathcal{D}'$ implies that $\alpha \in \mathcal{D}$, using the fact that $\tau_{min} \leq h_{min}$, $\tau_{max} \leq h_{max}$. Also, $\beta \in \mathcal{C}'$ implies that $\underline{\tau} \leq \overline{\tau}$ and $\underline{h} \leq \overline{h}$. Moreover, $\overline{\tau} = \min(-\beta_2, -\beta_4) \leq -\beta_4 = \overline{h}$. Hence, $\alpha \in \mathcal{C}$. Thus, $\alpha \in \mathcal{C} \cap \mathcal{D}$. Moreover, if $\beta \in \mathcal{C}'_o$ then $\beta \in \mathcal{C}' \cap \mathcal{D}'$ and $\mathsf{Stab}(f(\beta))$ gives $\alpha \in \mathcal{C} \cap \mathcal{D}$ and $\mathsf{Stab}(\alpha)$. Thus, $\alpha \in \mathcal{C}_o$. We now show that $\mathcal{C} \cap \mathcal{D} \subseteq f(\mathcal{C}' \cap \mathcal{D}')$ and $\mathcal{C}_o \subseteq f(\mathcal{C}'_o)$. Let $\alpha = (\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}) \in \mathcal{C} \cap \mathcal{D}$ and let $\beta = (\underline{\tau}, -\overline{\tau}, \underline{h}, -\overline{h})$. Then, $f(\beta) = (\underline{\tau}, \min(\overline{\tau}, \overline{h}), \underline{h}, \overline{h})$. Since $\alpha \in \mathcal{C}$, it follows that $\min(\overline{\tau}, \overline{h}) = \overline{\tau}$ and $f(\beta) = \alpha$. Moreover, it is straightforward to verify that $\alpha \in \mathcal{C} \cap \mathcal{D}$ implies $\beta \in \mathcal{C}' \cap \mathcal{D}'$ and that $\alpha \in \mathcal{C}_o$ implies $\beta \in \mathcal{C}'_o$. □

The previous result has two important implications. The first one is that the proposed re-parametrization does not introduce any conservatism in the solution to Problem 2 since the set $\mathcal{C}_o$ of admissible parameters α can be obtained by computing the set $\mathcal{C}'_o$ of admissible parameters β, despite the fact that the map f is not injective nor surjective. The second one is stated in the following lemma:

LEMMA 3. *Let $\mathcal{C}'^* \subseteq \mathcal{C}'_o$, then $\mathcal{C}^* = f(\mathcal{C}'^*)$ is a solution to Problem 2.*

PROOF. It holds that $\mathcal{C}^* = f(\mathcal{C}'^*) \subseteq f(\mathcal{C}'_o) = \mathcal{C}_o$. □

We further define the following set

$$\mathcal{E}'_o = \left\{\beta \in \mathcal{D}' : (\beta \notin \mathcal{C}') \vee ((\beta \in \mathcal{C}') \wedge \mathsf{Stab}(f(\beta)))\right\}$$

One can easily check that the following relation holds:

$$\mathcal{C}'_o = \mathcal{C}' \cap \mathcal{E}'_o. \tag{14}$$

Hence, from the previous equality and Lemma 3, we can solve Problem 2 by computing (a subset of) the set $\mathcal{E}'_o$. Moreover, $\mathcal{E}'_o$ satisfies the following monotonicity property:

PROPOSITION 1. *For all $\beta, \beta' \in \mathcal{D}'$, the following implications hold:*

$$\big((\beta \leq \beta') \wedge (\beta \in \mathcal{E}'_o)\big) \implies \beta' \in \mathcal{E}'_o.$$

$$\big((\beta \leq \beta') \wedge (\beta' \notin \mathcal{E}'_o)\big) \implies \beta \notin \mathcal{E}'_o.$$

PROOF. Let us assume $\beta \leq \beta'$ and $\beta \in \mathcal{E}'_o$. There are two cases:

1. If $\beta \notin \mathcal{C}'$, then either $-\beta'_4 \leq -\beta_4 < \beta_3 \leq \beta'_3$, or $-\beta'_2 \leq -\beta_2 < \beta_1 \leq \beta'_1$, or $-\beta'_4 \leq -\beta_4 < \beta_1 \leq \beta'_1$. In all three cases $\beta' \notin \mathcal{C}'$ and therefore $\beta' \in \mathcal{E}'_o$.
2. If $\beta \in \mathcal{C}'$ and $\mathsf{Stab}(f(\beta))$, then either $\beta' \notin \mathcal{C}'$ which implies $\beta' \in \mathcal{E}'_o$, or $\beta' \in \mathcal{C}'$. In this latter case, $\alpha = (\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}) = f(\beta)$ and $\alpha' = (\underline{\tau}', \overline{\tau}', \underline{h}', \overline{h}') = f(\beta')$ satisfy $\alpha \in \mathcal{C}$, $\alpha' \in \mathcal{C}$ and
$$\underline{\tau}' \geq \underline{\tau},\ \overline{\tau}' \leq \overline{\tau},\ \underline{h}' \geq \underline{h},\ \overline{h}' \leq \overline{h}. \tag{15}$$
It is straightforward to check that if (2-3) is GUES for $(\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}) \in \mathcal{C}$ then (2-3) is GUES for all $(\underline{\tau}', \overline{\tau}', \underline{h}', \overline{h}') \in \mathcal{C}$ satisfying (15). Thus, $\mathsf{Stab}(f(\beta'))$ holds and $\beta' \in \mathcal{E}'_o$.

This proves the first implication. For the second implication, it is sufficient to check that

$$\begin{array}{rl} & \big((\beta \leq \beta') \wedge (\beta \in \mathcal{E}'_o)\big) \implies \beta' \in \mathcal{E}'_o \\ \equiv & \neg(\beta \leq \beta') \vee (\beta \notin \mathcal{E}'_o) \vee (\beta' \in \mathcal{E}'_o) \\ \equiv & \big((\beta \leq \beta') \wedge (\beta' \notin \mathcal{E}'_o)\big) \implies \beta \notin \mathcal{E}'_o. \end{array}$$

□

The previous property is instrumental for computing a subset of $\mathcal{E}'_o$ since it allows us to state the following theorem:

THEOREM 4. *Let $\underline{\beta}^1, \ldots, \underline{\beta}^{M_1} \in \mathcal{E}'_o$, and $\overline{\beta}^1, \ldots, \overline{\beta}^{M_2} \in \mathcal{D}' \setminus \mathcal{E}'_o$ and let*

$$\underline{\mathcal{E}}' = \bigcup_{j=1}^{M_1} \{\beta \in \mathcal{D}' : \underline{\beta}^j \leq \beta\},\ \overline{\mathcal{E}}' = D' \setminus \bigcup_{j=1}^{M_2} \{\beta \in \mathcal{D}' : \beta \leq \overline{\beta}^j\}.$$

Then, $\underline{\mathcal{E}}' \subseteq \mathcal{E}'_o \subseteq \overline{\mathcal{E}}'$. Moreover, $\mathcal{C}^* = f(\mathcal{C}' \cap \underline{\mathcal{E}}')$ is a solution to Problem 2 and $\mathcal{C}_o \subseteq f(\mathcal{C}' \cap \overline{\mathcal{E}}')$.

PROOF. $\underline{\mathcal{E}}' \subseteq \mathcal{E}'_o \subseteq \overline{\mathcal{E}}'$ is a direct consequence of Proposition 1. Then, from (14) and Lemmas 2 and 3, it follows that $\mathcal{C}^*$ is a solution to Problem 2 and $\mathcal{C}_o \subseteq f(\mathcal{C}' \cap \overline{\mathcal{E}}')$. □

5.2 Timing contract synthesis algorithm

The previous theorem shows that it is possible to compute under and over-approximations of the set $\mathcal{E}'_o$ by sampling the parameter space $\mathcal{D}'$. In this section, we use this property to design a synthesis algorithm. Similar algorithms have been used in [25, 30] for computing an approximation

of the Pareto front of a monotone multi-criteria optimization problem. Indeed, this latter problem can be tackled by computing an under and over-approximation of a set satisfying a monotonicity property similar to that of Proposition 1.

ALGORITHM 2. *Timing contract synthesis*

function *TC_Synth(A_c,A_a,A_s,$\mathcal{D}$)*
input: A_c, A_a, $A_s \in \mathbb{R}^{n\times n}$, $\mathcal{D} = [\tau_{min}, \tau_{max}]^2 \times [h_{min}, h_{max}]^2$
output: $\mathcal{C}^* \subseteq \mathcal{C} \cap \mathcal{D}$ *such that for all* $(\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}) \in \mathcal{C}^*$, *(2-3) is GUES.*
parameter: $\varepsilon \in \mathbb{R}^+$

1: ***if*** $\beta_{min} = (\tau_{min}, -\tau_{max}, h_{min}, -h_{max}) \in \mathcal{E}'_o$ ***then***
2: *return* $\mathcal{C} \cap \mathcal{D}$;
3: ***else*** $\overline{\mathcal{E}}' := \mathcal{D}' \setminus \{\beta_{min}\}$;
4: ***end if***
5: ***if*** $\beta_{max} = (\tau_{max}, -\tau_{min}, h_{max}, -h_{min}) \notin \mathcal{E}'_o$ ***then***
6: *return* $\emptyset$;
7: ***else*** $\underline{\mathcal{E}}' := \{\beta_{max}\}$;
8: ***end if***
9: ***while*** $d(\underline{\mathcal{E}}', \overline{\mathcal{E}}') > \varepsilon$ ***do*** ▷ *main loop*
10: *Pick* $\beta \in \overline{\mathcal{E}}' \setminus \underline{\mathcal{E}}'$; ▷ *select next sample*
11: ***if*** $\beta \in \mathcal{E}'_o$ ***then*** $\underline{\mathcal{E}}' := \underline{\mathcal{E}}' \cup \{\beta' \in \mathcal{D}' : \beta \leq \beta'\}$;
12: ***else*** $\overline{\mathcal{E}}' := \overline{\mathcal{E}}' \setminus \{\beta' \in \mathcal{D}' : \beta' \leq \beta\}$;
13: ***end if***
14: ***end while***
15: *return* $f(\mathcal{C}' \cap \underline{\mathcal{E}}')$;

Algorithm 2 computes an under-approximation $\underline{\mathcal{E}}'$ and an over-approximation $\overline{\mathcal{E}}'$ of the set $\mathcal{E}'_o$ by sampling iteratively the parameter space $\mathcal{D}'$.

Lines 1 to 8 correspond to the initialization of these approximations by testing the lower bound β_{min} and the upper bound β_{max} of the set $\mathcal{D}'$. If $\beta_{min} \in \mathcal{E}'_o$, then by Theorem 4, $f(\mathcal{C}' \cap \mathcal{D}') = \mathcal{C} \cap \mathcal{D}$ is a solution to Problem 2. Note that in that case, all timing-contract parameters in $\mathcal{C} \cap \mathcal{D}$ guarantee the stability of (2-3). If $\beta_{min} \notin \mathcal{E}'_o$, then $\mathcal{D}' \setminus \{\beta_{min}\}$ is an over-approximation of $\mathcal{E}'_o$. Similarly, if $\beta_{max} \notin \mathcal{E}'_o$, then by Theorem 4, $\mathcal{E}'_o = \emptyset$. Note that in that case, no timing-contract parameters in $\mathcal{C} \cap \mathcal{D}$ can guarantee the stability of (2-3). If $\beta_{max} \in \mathcal{E}'_o$, then $\{\beta_{max}\}$ is an under-approximation of $\mathcal{E}'_o$.

Lines 9 to 14 describe the main loop of the timing contract synthesis algorithm. At any time of the execution, $\underline{\mathcal{E}}' \subseteq \mathcal{E}'_o \subseteq \overline{\mathcal{E}}'$ holds. We pick a sample $\beta \in \overline{\mathcal{E}}' \setminus \underline{\mathcal{E}}'$ which is the unexplored parameter region lying in the over-approximation of $\mathcal{E}'_o$ but not in its under-approximation. If $\beta \in \mathcal{E}'_o$ (or if $\beta \notin \mathcal{E}'_o$), then we update the under-approximation $\underline{\mathcal{E}}'$ (or the over-approximation $\overline{\mathcal{E}}'$) according to Theorem 4. The algorithm stops when the Hausdorff distance between the $\underline{\mathcal{E}}'$ and $\overline{\mathcal{E}}'$ becomes smaller than ε. Of course, the choice of the sample $\beta \in \overline{\mathcal{E}}' \setminus \underline{\mathcal{E}}'$, at line 10, is crucial for the efficiency of the algorithm. In our implementation of the algorithm, we use the selection criteria proposed in [25] which consists in choosing the sample that will produce the fastest decrease of the Hausdorff distance $d(\underline{\mathcal{E}}', \overline{\mathcal{E}}')$. In [30] an alternative selection criteria based on multiscale grid exploration was proposed.

Finally, it is important to note that Algorithm 2 needs testing if the samples $\beta \in \mathcal{E}'_o$ which require checking the condition $\mathsf{Stab}(f(\beta))$. In our implementation, this is done using Algorithm 1. If it returns `true`, then we can consider that $\mathsf{Stab}(f(\beta))$ holds. If it returns `unknown`, we treat the sample as if $\mathsf{Stab}(f(\beta))$ is false. As a consequence, in practice it may be the case that $\overline{\mathcal{E}}'$ is not an over-approximation of $\mathcal{E}'_o$. However, it always holds that $\underline{\mathcal{E}}' \subseteq \mathcal{E}'_o$ and therefore the set returned by Algorithm 2 is always a valid solution to Problem 2. Note that the property $\mathsf{Stab}(f(\beta))$ need not be checked using Algorithm 1 but one can use any of the algorithms mentioned in Table 1.

6. ILLUSTRATIVE EXAMPLES

In this section, we first compare our approach for stability verification to that implemented within the NCS toolbox [5]. Then, we show an application of the timing contract synthesis algorithm. We implemented Algorithm 1 and Algorithm 2 in Matlab using the Multi-Parametric Toolbox [19]. All reported experiments are realized on a desktop with i7 4790 processor of frequency 3.6 GHz and a 8 GB RAM.

6.1 Stability Verification

We consider two systems taken from [9], given by (1) with the following matrices:

$$A = \begin{pmatrix} 0 & 1 \\ 0 & -0.1 \end{pmatrix},\ B = \begin{pmatrix} 0 \\ 0.1 \end{pmatrix},\ K = \begin{pmatrix} -3.75 & -11.5 \end{pmatrix}. \tag{16}$$

$$A = \begin{pmatrix} 0 & 1 \\ -2 & 0.1 \end{pmatrix},\ B = \begin{pmatrix} 0 \\ 1 \end{pmatrix},\ K = \begin{pmatrix} 1 & 0 \end{pmatrix}. \tag{17}$$

We consider the stability verification problem for these two 2-dimensional systems. First, we write the systems into 4-dimensional impulsive systems (3). Then, we apply Algorithm 1 to check stability of the impulsive system under several timing contracts. We compare our results to those obtained using the NCS toolbox [5] in Table 2. For the DET timing contract ($\underline{\tau} = 0$, $\underline{h} = \overline{h} = h$), we fix parameter h and report the maximal value of $\overline{\tau}$ for which stability has been verified. For the timing contract that corresponds to NPILS ($\underline{\tau} = \overline{\tau} = 0$), we fix $\underline{h}$ and report the maximal value of $\overline{h}$ for which stability has been verified. Finally, for the general timing contract given by (2), we fix parameters $\underline{\tau}$, $\overline{\tau}$, $\underline{h}$ and report the maximal value of $\overline{h}$ for which stability has been verified. Note that we conducted extra experiments labelled "Algorithm 1 (exp1)" to compare the results in terms of CPU time after fixing the same parameters as those used with the NCS toolbox.

The experiments conducted using the NCS toolbox are done in a particular manner since it uses three different approximation methods to embed the timing uncertainty (Jordan Normal Form (JNF), Cayley Hamilton, and Gridding and Norm Bounding (GNB)): we search for the maximum value of the free timing parameter that guarantees stability by running experiments using the three approximation methods. Then we report the computation time for the experiment in which we obtained this bound. In case the maximum bound could be obtained by more than one experiment, we report the CPU time corresponding to the fastest in terms of computation. Stability for system (16) is guaranteed using the GNB approximation for the DET, NPILS and general contracts, with 50, 35, and 50 gridpoints respectively. As for system (17), stability is guaranteed using the JNF approximation for all three contracts. Parameter setups used by Algorithm 1, for the different experiments, are summarized by Table 3. Note that for the NPILS contract,

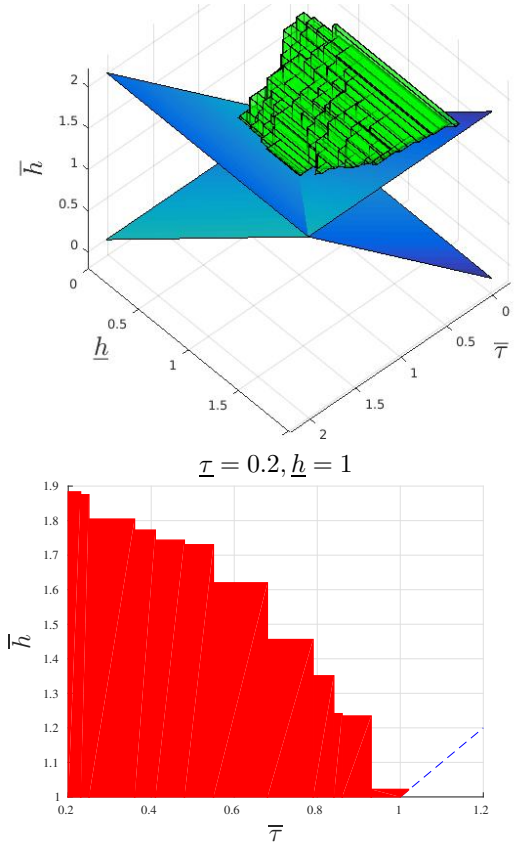

Figure 3: (top) Timing contract synthesis for system (17) in the $(0.2, \overline{\tau}, \underline{h}, \overline{h})$ space where the visualized section of $\mathcal{C}^*$ is in the domain region $\mathcal{C}$ defined above the planes $\overline{h} \geq \overline{\tau}$ and $\overline{h} \geq \underline{h}$. (bottom) The section of $\mathcal{C}^*$ in the $(\overline{\tau}, \overline{h})$ plane such that $\overline{\tau} = 0.2$ and $\underline{h} = 1$.

the parameter N_1 has no effect. It is clear, for the two systems at hand, that our method gives better results than the NCS toolbox in terms of CPU time and tightness.

6.2 Contract Synthesis

We now consider the timing contract synthesis problem for system (17). We rewrite the system in the form of impulsive system (3). We search for a set $\mathcal{C}^* \subseteq \mathcal{C} \cap \mathcal{D}$, where $\mathcal{D} = [0, 1.16]^2 \times [0.21, 2.02]^2$, such that for all $(\underline{\tau}, \overline{\tau}, \underline{h}, \overline{h}) \in \mathcal{C}^*$ the system (2-3) is guaranteed to be GUES. We set the parameter $\varepsilon = 0.07$, and apply Algorithm 2. The output of the latter is a set $\mathcal{C}^* \subset \mathbb{R}^4$. Figure 3 shows a 3D section of $\mathcal{C}^*$ by setting $\underline{\tau} = 0.2$, and a 2D section by setting $\underline{\tau} = 0.2$ and $\underline{h} = 1$. Algorithm 2 used 3094 samples in the 4 dimensional parameter space with a total computation time of $T_{CPU} = 250$ minutes. Parameters of Algorithm 1 used in Algorithm 2 are $L = 4$, $k_{max} = 5$ and the numbers of time steps used for the over-approximation of the reachable set are $N_1 = 20$ and $N_2 = 50$.

7. CONCLUSION

In this work, we proposed useful tools for contract-based design of embedded control systems under the form of algorithms for stability verification and timing contract synthesis. These algorithms can be used by control and software engineers to derive requirements that must be met by the real-time implementation of a control law. The effectiveness of our approach has been shown on examples. As future work, it would be interesting to handle the problem of controller synthesis given a timing contract, and to co-synthesize the controller and the timing contract parameters. Also more work is required for the stability verification problem as long as the solutions at hand gives only sufficient conditions.

References

[1] M. Al Khatib, A. Girard, and T. Dang. Stability verification of nearly periodic impulsive linear systems using reachability analysis. In *IFAC Conference on Analysis and Design of Hybrid Systems*, pages 358–363, 2015.

[2] M. Althoff, O. Stursberg, and M. Buss. Computing reachable sets of hybrid systems using a combination of zonotopes and polytopes. *Nonlinear Analysis: Hybrid Systems*, 4(2):233–249, 2010.

[3] N. Athanasopoulos and M. Lazar. Alternative stability conditions for switched discrete time linear systems. In *IFAC World Congress*, pages 6007–6012, 2014.

[4] N. W. Bauer, P. J. H. Maas, and W. P. M. H. Heemels. Stability analysis of networked control systems: A sum of squares approach. *Automatica*, 48(8):1514–1524, 2012.

[5] N. W. Bauer, S. J. L. M. van Loon, M. C. F. Donkers, N. van de Wouw, and W. P. M. H. Heemels. Networked control systems toolbox: Robust stability analysis made easy. In *IFAC Workshop on Distributed Estimation and Control in Networked Systems*, pages 55–60, 2012.

[6] F. Blanchini. Ultimate boundedness control for uncertain discrete-time systems via set-induced lyapunov functions. In *IEEE Conference on Decision and Control*, pages 1755–1760, 1991.

[7] O. Botchkarev and S. Tripakis. Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. In *Hybrid Systems: Computation and Control*, pages 73–88. Springer, 2000.

[8] M. C. Bragagnolo, I.-C. Morarescu, J. Daafouz, and P. Riedinger. LMI sufficient conditions for the consensus of linear agents with nearly-periodic resets. In *American Control Conference*, pages 2575–2580. IEEE, 2014.

[9] C. Briat. Convex conditions for robust stability analysis and stabilization of linear aperiodic impulsive and sampled-data systems under dwell-time constraints. *Automatica*, 49(11):3449–3457, 2013.

[10] M. B. G. Cloosterman, L. Hetel, N. Van De Wouw, W. P. M. H. Heemels, J. Daafouz, and H. Nijmeijer. Controller synthesis for networked control systems. *Automatica*, 46(10):1584–1594, 2010.

[11] P. Derler, E. A. Lee, S. Tripakis, and M. Törngren. Cyber-physical system design contracts. In *ACM/IEEE International Conference on Cyber-Physical Systems*, pages 109–118, 2013.

Table 2: Results of Algorithm 1 for systems (16) and (17) under several timing contracts. T_{CPU} is the computation time in seconds.

		DET ($\underline{\tau}=0$, $\underline{h}=\overline{h}=h$)			NPILS ($\underline{\tau}=\overline{\tau}=0$)			General contract (2)				
		$\overline{\tau}$	h	T_{CPU}	$\underline{h}$	$\overline{h}$	T_{CPU}	$\underline{\tau}$	$\overline{\tau}$	$\underline{h}$	$\overline{h}$	T_{CPU}
System (16)	NCS toolbox (GNB)	0.63	1	3.42	10^{-3}	1.7291	3.30	0	0.4	0.2	1.13	9.17
	Algorithm 1(exp1)	0.63	1	0.18	10^{-3}	1.7291	0.20	0	0.4	0.2	1.13	4.49
	Algorithm 1(exp2)	0.67	1	1.16	10^{-3}	1.7294	0.20	0	0.4	0.2	1.23	9.95
System (17)	NCS toolbox (JNF)	0.78	1	2.07	0.4	0.45	1.91	0	0.1	0.4	0.44	3.62
	Algorithm 1(exp1)	0.78	1	0.41	0.4	0.45	0.21	0	0.1	0.4	1.71	1.13
	Algorithm 1(exp2)	1	1	2.97	0.4	1.88	1.22	0	0.1	0.4	1.71	5.15

Table 3: Parameter setup for Algorithm 1 for systems (16) and (17) under several timing contracts.

	DET ($\underline{\tau}=0$, $\underline{h}=\overline{h}=h$)				NPILS ($\underline{\tau}=\overline{\tau}=0$)			General contract (2)			
	k_{max}	N_1	N_2	L	k_{max}	N_2	L	k_{max}	N_1	N_2	L
System (16)(exp1)	30	30	1	2	30	100	2	30	10	10	4
System (16)(exp2)	30	100	1	2	30	1000	2	30	20	50	4
System (17)(exp1)	30	15	1	2	30	1	2	30	10	1	4
System (17)(exp2)	30	150	1	2	30	100	2	30	20	60	4

[12] M. C. F. Donkers, W. P. M. H. Heemels, N. Van De Wouw, and L. Hetel. Stability analysis of networked control systems using a switched linear systems approach. *IEEE Transactions on Automatic Control*, 56(9):2101–2115, 2011.

[13] M. Fiacchini and I.-C. Morarescu. Set theory conditions for stability of linear impulsive systems. In *IEEE Conference on Decision and Control*, pages 1527–1532, 2014.

[14] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. SpaceEx: Scalable verification of hybrid systems. In *Computer Aided Verification*, pages 379–395. Springer, 2011.

[15] H. Fujioka. Stability analysis of systems with aperiodic sample-and-hold devices. *Automatica*, 45(3):771–775, 2009.

[16] H. Gao, X. Meng, T. Chen, and J. Lam. Stabilization of networked control systems via dynamic output-feedback controllers. *SIAM Journal on Control and Optimization*, 48(5):3643–3658, 2010.

[17] A. Girard. Reachability of uncertain linear systems using zonotopes. In *Hybrid Systems: Computation and Control*, pages 291–305. Springer, 2005.

[18] W. P. M. H. Heemels, A. R. Teel, N. Van de Wouw, and D. Nešić. Networked control systems with communication constraints: Tradeoffs between transmission intervals, delays and performance. *IEEE Transactions on Automatic Control*, 55(8):1781–1796, 2010.

[19] M. Herceg, M. Kvasnica, C. Jones, and M. Morari. Multi-Parametric Toolbox 3.0. In *European Control Conference*, pages 502–510, 2013.

[20] L. Hetel, J. Daafouz, S. Tarbouriech, and C. Prieur. Stabilization of linear impulsive systems through a nearly-periodic reset. *Nonlinear Analysis: Hybrid Systems*, 7(1):4–15, 2013.

[21] L. Hetel, A. Kruszewski, W. Perruquetti, and J.-P. Richard. Discrete and intersample analysis of systems with aperiodic sampling. *IEEE Transactions on Automatic Control*, 56(7):1696–1701, 2011.

[22] A. B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability analysis: internal approximation. *Systems & Control Letters*, 41(3):201–211, 2000.

[23] C. Le Guernic. *Reachability analysis of hybrid systems with linear continuous dynamics*. PhD thesis, Université Joseph-Fourier-Grenoble I, 2009.

[24] C. Le Guernic and A. Girard. Reachability analysis of linear systems using support functions. *Nonlinear Analysis: Hybrid Systems*, 4(2):250–262, 2010.

[25] J. Legriel, C. Le Guernic, S. Cotton, and O. Maler. Approximating the pareto front of multi-criteria optimization problems. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 69–83. Springer, 2010.

[26] K. Liu, E. Fridman, and L. Hetel. Networked control systems in the presence of scheduling protocols and communication delays. *SIAM Journal on Control and Optimization*, 53(4):1768–1788, 2015.

[27] K. Liu, V. Suplin, and E. Fridman. Stability of linear systems with general sawtooth delay. *IMA Journal of Mathematical Control and Information*, 27(4):419–436, 2010.

[28] A. Sangiovanni-Vincentelli, W. Damm, and R. Passerone. Taming dr. frankenstein: Contract-based design for cyber-physical systems. *European journal of control*, 18(3):217–238, 2012.

[29] A. Seuret and M. Peet. Stability analysis of sampled-data systems using sum of squares. *IEEE Transactions on Automatic Control*, 58(6):1620–1625, 2013.

[30] P. Tendulkar. *Mapping and Scheduling on Multi-core Processors using SMT Solvers*. PhD thesis, Universite de Grenoble I-Joseph Fourier, 2014.